



# Applying Trajectory approach with static priority queuing for improving the use of available AFDX resources

Henri Bauer, Jean-Luc Scharbarg, Christian Fraboul

## ► To cite this version:

Henri Bauer, Jean-Luc Scharbarg, Christian Fraboul. Applying Trajectory approach with static priority queuing for improving the use of available AFDX resources. 18th International Conference on Real-Time and Network Systems, Nov 2010, Toulouse, France. pp.69-78. hal-00544508

**HAL Id: hal-00544508**

**<https://hal.science/hal-00544508>**

Submitted on 8 Dec 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Applying Trajectory approach with static priority queuing for improving the use of available AFDX resources

Henri Bauer<sup>1,2</sup>, Jean-Luc Scharbarg<sup>2</sup>, Christian Fraboul<sup>2</sup>

<sup>1</sup> Airbus France - 316 Route de Bayonne 31300 Toulouse, France

<sup>2</sup> Université de Toulouse - IRIT-ENSEEIH-INT - 2, rue Camichel 31000 Toulouse, France  
{Henri.Bauer, Jean-Luc.Scharbarg, Christian.Fraboul}@enseeiht.fr

## Abstract

*AFDX (Avionics Full Duplex Switched Ethernet) standardized as ARINC 664 is a major upgrade for avionics systems. The mandatory certification implies a worst-case delay analysis of all the flows transmitted on the AFDX network. Up to now, this analysis is done thanks to a tool based on a Network Calculus approach. The more recent Trajectory approach has been proposed for the computation of worst-case response time in distributed systems. This paper shows how the worst-case delay analysis of an AFDX network can be improved using an optimized Trajectory approach. This approach, which implements static priority QoS policies, provides bound needed for deterministic avionics flows (high priority) and allows addition of (lower priority) non avionics flows.*

## 1 Introduction

Designing and manufacturing new civilian aircraft has lead to an increase of the number of embedded systems and functions. The AFDX [1] brings an answer by multiplexing huge amount of communication flows over a full duplex switched Ethernet network. It has become the reference communication technology in the context of civilian avionics and provides a backbone network for the avionics platform.

Full duplex switched Ethernet eliminates the inherent indeterminism of vintage (CSMA-CD) Ethernet. Nevertheless, it shifts the indeterminism problem to the switch level where various flows can enter in competition for sharing output ports of a given switch.

Main AFDX specific assumptions deal with the static definition of avionics flows which are described as multicast links. All the flows are asynchronous, but have to respect a bandwidth envelope (burst and rate) at network ingress point. Each flow is statically mapped on the network of interconnected AFDX switches. These specific assumptions allow end-to-end delay analysis of each flow of a given avionics configuration mapped on a given network of interconnected AFDX switches.

For a given flow, the end-to-end communication delay of a packet is the sum of transmission delays on links and latencies in switches. As the links are full duplex there is no packet collision on links [6]. The transmission delay only depends on the transmission rate and on the packet length. But, the latency in switches is highly variable because of the confluence of asynchronous flows, which compete on each switch output port (according to servicing policy). Therefore, it is necessary to analyze precisely the latency in every switch output port in order to determine upper bounds on end-to-end delay and jitter of each flow [3, 2].

Previous work has been devoted to the worst case analysis of end-to-end delays on an AFDX network.

For certification reasons, a first tool, based on the Network Calculus theory and implemented by Rockwell Collins, has been proposed for the computation of an upper bound for the end-to-end delay of each flow. This approach models the traffic on the AFDX network as a set of sporadic flows with no QoS classes differentiation. The input flows and the output ports are respectively modeled with traffic envelopes and service curves. Since these envelopes and curves are pessimistic, the obtained upper bounds are pessimistic. The Network Calculus approach has been improved in the context of AFDX by adding a grouping technique (flows sharing a common link are serialized and cannot arrive at the same time on a switch) [4, 5].

The model-checking approach presented in [3] computes the exact worst-case delay of each flow. Unfortunately, it cannot cope with real AFDX configurations, due to the combinatorial explosion problem for large configurations. Nevertheless, it is used in this paper as a reference for exact worst-case computation on an illustrative small configuration.

This paper deals with a third approach [9] which is based on the Trajectory concept. It identifies for a packet  $m$  the busy periods and the packets impacting its end-to-end delay on all the nodes visited by  $m$ . Thus, it allows a worst-case delay computation. This approach has been applied [2] to AFDX in the case of a FIFO output port policy. In this paper, we use the Trajectory approach with

fixed priority policy in order to provide the bounds needed for a deterministic avionics network with a static priority QoS policy. The idea is to introduce additional non avionics traffic (with lower priority) for improving the use of available AFDX resources.

A first contribution of this paper is to present how existing results for worst case response time of flows scheduled with a combined Fixed Priority (FP) and First In, First Out (FIFO) algorithm [9] can be applied to QoS AFDX worst case delay analysis.

A second contribution of this paper deals with the explanation of how the FP/FIFO Trajectory approach can be optimized by introducing the serialization of flows (similar to the grouping technique proposed in the Network Calculus context) with fixed priorities.

The paper is organized as follows. Section 2 shortly introduces the AFDX worst case delay analysis context. In Section 3, we explain how the trajectory approach can be employed to analyze end-to-end communication delays on a network with differentiated QoS traffic classes. Section 4 illustrates the approach on a representative part of an AFDX network.

## 2 The AFDX network worst case delay analysis

The AFDX is a switched Ethernet network taking into account avionics constraints. An illustrative example is depicted in Figure 1. It is composed of five interconnected switches  $S1$  to  $S5$ . Each switch has no input buffers on input ports and one FIFO buffer for each output port. The inputs and outputs of the network are called *End Systems* ( $e1$  to  $e10$  in Figure 1). Each end system is connected to exactly one switch port and each switch port is connected to at most one end system. Links between switches are all full duplex.

The end-to-end avionics traffic characterization is made by the definition of *Virtual Links*. As standardized by ARINC-664, Virtual Link (VL) is a concept of virtual communication channel. Thus it is possible to statically define all the flows (VL) which enter the network [1].

End systems exchange packets through VLs. Switching a packet from a transmitting to a receiving end system is based on VL. The Virtual Link defines a logical unidirectional connection from one source end system to one or more destination end systems. Coming back to the example in Figure 1,  $vx$  is a unicast VL with path  $\{e3 - S3 - S4 - e8\}$ , while  $v6$  is a multicast VL with paths  $\{e1 - S1 - S2 - e7\}$  and  $\{e1 - S1 - S4 - e8\}$ .

The routing is statically defined. Only one end system within the avionics network can be the source of one Virtual Link, (ie. mono transmitter assumption). A VL definition also includes the Bandwidth Allocation Gap ( $BAG$ ), the minimum and the maximum packet length ( $s_{min}$  and  $s_{max}$ ).  $BAG$  is the minimum delay between two consecutive packets of the associated VL (which actually defines a VL as a sporadic flow).

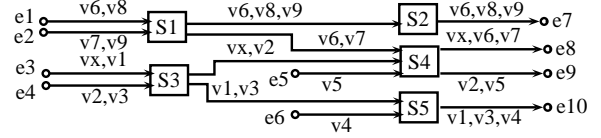


Figure 1. An illustrative AFDX configuration

VL parameters ( $BAG$ ,  $s_{max}$ ) compliance is ensured by a shaping unit at end system level and a traffic policing unit at each switch entry port (specificity of AFDX switches, compared to standard Ethernet switches). The delay incurred by the switching fabric is upper bounded by a constant value, i.e.  $16 \mu s$ .

All these constraints that the AFDX model adds to the vintage Ethernet enables a precise analysis of the network, especially the computation of an upper bound for the end-to-end delay of each flow and the dimensioning of output buffers so that no packet is lost.

The next step is to introduce additional load (lower priority non avionics flows) but to guaranty that existing avionics flows remain fully deterministic. We consider, in order to better use the network resources, that the deterministic constraint for additional traffic is less critical. We then have two distinct classes of flows whose properties are statically defined. These properties correspond well to a fixed priority servicing policy.

In the next section we consider a delay analysis method that provides deterministic bounds for the existing avionics flows and runs a QoS policy taking into account static priorities.

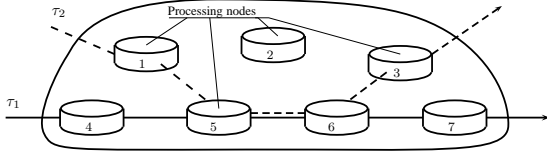
## 3 Trajectory approach on AFDX flows scheduled with FP/FIFO

The Trajectory approach [7, 8] has been developed to get deterministic upper bounds on end-to-end response time in distributed systems. This approach considers a set of sporadic flows with no assumption concerning the arrival time of packets. The principle of the application of the Trajectory approach to the AFDX has been presented in [2]. Main features of the Trajectory approach applied to AFDX are summarized and illustrated in Sections 3.1 and 3.2. The optimization of the Trajectory approach computation is presented in Section 3.3.

### 3.1 The main features of the Trajectory approach

The approach developed for the analysis of the AFDX considers the results from [9]. The general architecture of the distributed system considered in [9] is depicted in Figure 2.

Such a system is composed of a set of interconnected processing nodes (seven in Figure 2). Each flow crossing this system follows a static path which is an ordered sequence of nodes. In the example of Figure 2, there are two flows  $\tau_1$  and  $\tau_2$ .  $\tau_1$  follows the path  $\mathcal{P}_1 = \{4, 5, 6, 7\}$ .



**Figure 2. A distributed system**

Node 4 is the ingress node of  $\tau_1$  in the system. The Trajectory approach assumes, with regards to any flow  $\tau_i$  following path  $\mathcal{P}_i$ , that any flow  $\tau_j$  following path  $\mathcal{P}_j$ , with  $\mathcal{P}_j \neq \mathcal{P}_i$  and  $\mathcal{P}_j \cap \mathcal{P}_i \neq \emptyset$ , never visits a node of path  $\mathcal{P}_i$  after having left this path. In the example of Figure 2,  $\mathcal{P}_2 = \{1, 5, 6, 3\}$  and  $\mathcal{P}_1 \cap \mathcal{P}_2 = \{5, 6\}$ .

Each flow  $\tau_i$  has a minimum inter-arrival time between two consecutive packets at ingress node, denoted  $T_i$ , a maximum release jitter at the ingress node denoted  $J_i$ , a fixed priority level denoted  $P_i$ , an end-to-end deadline  $D_i$  that is the maximum end-to-end response time acceptable and a maximum processing time  $C_i^h$  on each node  $N_h$ , with  $N_h \in \mathcal{P}_i$ .

Flows are scheduled with a combined Fixed Priority (FP) and First In, First Out (FIFO) algorithm in every visited node (non preemptive policy). The flows are at first sorted according to their fixed priority level, and flows with same fixed priority are then treated in FIFO order. For each flow  $\tau_i$ , we define three sets:

- $hp_i = \{j \in [1, n], P_j > P_i\}$ , the set of flows having a fixed priority strictly higher than this of flow  $\tau_i$ ;
- $sp_i = \{j \in [1, n], j \neq i, P_j = P_i\}$ , the set of flows distinct of  $\tau_i$  having a fixed priority equal to this of flow  $\tau_i$ ;
- $lp_i = \{j \in [1, n], P_j < P_i\}$ , the set of flows having a fixed priority strictly lower than this of flow  $\tau_i$ ,

The transmission time of any packet on any link between nodes has known lower and upper bounds  $L_{min}$  and  $L_{max}$  and there are neither collisions nor packet losses on links.

The end-to-end response time of a packet is the sum of the times spent in each crossed node and the transmission delays on links. The transmission delays on links are upper bounded by  $L_{max}$ . The time spent by a packet  $m$  in a node  $N_h$  depends on the higher priority packets in node  $N_h$  and on the delay due to the non preemption of at most one lower priority packet. The higher priority packets can be grouped into two categories. The first one contains the packets with the same fixed priority than packet  $m$  that have arrived in  $N_h$  before the arrival time of  $m$  in  $N_h$  (all these packets have a higher dynamic priority than  $m$ , considering the SP/FIFO scheduling, and thus, will be processed before  $m$ ). The other category includes the packets with a higher fixed priority than packet  $m$  that have arrived before  $m$  begins to be transmitted by  $N_h$ . The problem is then to upper bound the overall time spent in the visited nodes.

The solution proposed by the Trajectory approach is based on the busy period concept. A busy period of level  $\mathcal{L}$  is an interval  $[t, t')$  such that  $t$  and  $t'$  are both idle times of level  $\mathcal{L}$  and there is no idle time of level  $\mathcal{L}$  in  $(t, t')$ . An idle time  $t$  of level  $\mathcal{L}$  is a time such as all packets with priority greater than or equal to  $\mathcal{L}$  generated before  $t$  have been processed at time  $t$ .

The Trajectory approach considers a packet  $m$  from flow  $\tau_i$  generated at time  $t$ . It identifies the busy period and the packets impacting its end-to-end delay on all the nodes visited by  $m$  (starting from the last visited node backward to the ingress node). This decomposition enables the computation of the latest starting time of  $m$  on its last node. This starting time can be computed recursively and leads to the worst case end-to-end response time of the flow  $\tau_i$ . This computation will be illustrated in the context of AFDX.

The elements of the system considered in the Trajectory approach are instantiated in the following way in the context of AFDX:

- each node of the system corresponds to an AFDX switch output port, including the output link,
- each link of the system corresponds to the switching fabric,
- each flow corresponds to a VL path.

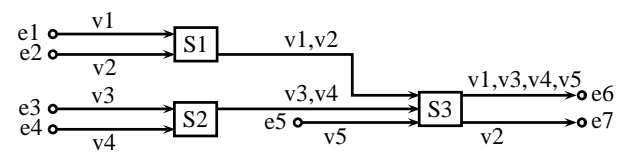
The assumptions of the Trajectory approach are verified by the AFDX. Indeed, switch output ports implement FP/FIFO service discipline. The switching fabric delay is upper bounded by a constant value ( $16\mu s$ ), thus  $L = L_{min} = L_{max} = 16\mu s$ . There are no collisions nor packet loss on AFDX networks. The routing of the VLs is statically defined.

VL parameters match the definition of sporadic flows in the following manner:  $T_i = BAG$ ,  $C_i^h = s_{max}/R$ ,  $J_i = 0$ . Since all the AFDX ports work at the same rate  $R = 100Mb/s$ ,  $C_i^h = C_i = s_{max}/R$  for every node  $h$  in the network.

### 3.2 Illustration on a sample AFDX configuration

#### 3.2.1 Identification of the worst-case for a packet

Let us consider a sample AFDX configuration depicted in Figure 3. The five VLs  $v1, \dots, v5$  which are transmitted on this AFDX network all have the same BAG ( $4000\mu s$ ) and the same  $s_{max}$  ( $4000$  bits). All VLs have the same priority level, except for  $v1$  which has a higher fixed priority level.



**Figure 3. Illustrative AFDX configuration**

Figure 4 shows an arbitrary scheduling of the packets, which are identified by their VL numbers (eg. packet 3 is a packet from VL  $v3$ ). The scheduling in Figure 4 focuses on packet 3. The arrival time of a packet  $m$  in a node  $N_h$  is denoted  $a_m^{N_h}$ . Time origin is arbitrarily chosen as the arrival time of packet 3 in node  $e3$ . The processing time of a packet in a node is  $40 \mu s$ . It corresponds to the transmission time of the packet on a link. The delay between the end of the processing of a packet by a node and its arrival in the next node corresponds to the  $16 \mu s$  switch factory delay. In each node, the packets are processed with respect to the SP/FIFO policy. Consequently, packet 3 is delayed by packet 4 in  $S2$ . In node  $S3$ , packet 5 is delayed by packet 1 and delays packet 4, which delays packet 3.

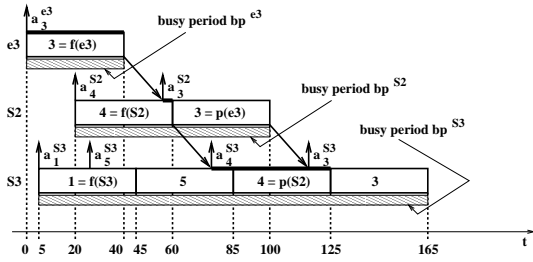


Figure 4. An arbitrary scheduling of packets

Packet 3 from VL  $v3$  crosses three busy periods ( $bp^{e3}$ ,  $bp^{S2}$ , and  $bp^{S3}$ ) on its trajectory, corresponding to the three nodes  $N_1 = e3$ ,  $N_2 = S2$  and  $N_3 = S3$ . Let  $f(N_i)$  be the first packet which is processed in the busy period  $bp^{N_i}$  during which packet 3 is processed. Considering the scheduling in Figure 4, we have  $f(e3) = 3$ ,  $f(S2) = 4$  and  $f(S3) = 1$ . As flows do not necessarily follow the same path in the network, it is possible that packet  $f(N_i)$  does not come from the same previous node  $N_{i-1}$  as packet 3. This case occurs in node  $S2$ , where packet 4 comes from node  $e4$ . It also occurs in node  $S3$ , where packet 1 comes from node  $S1$ . Therefore,  $p(N_{i-1})$  is defined as the first packet which is processed in  $bp^{N_i}$  and comes from node  $N_{i-1}$ . Considering the scheduling in Figure 4, we have  $p(e3) = 3$  and  $p(S2) = 4$ .

The starting time of packet 3 in node  $S3$  is obtained by adding parts of the three busy periods  $bp^{e3}$ ,  $bp^{S2}$ , and  $bp^{S3}$  to the delays between the nodes, i.e.  $2 \times 16 \mu s$ . From [9], the part of the busy period  $bp^{N_i}$  which has to be added is the processing time of packets between  $f(N_i)$  and  $p(N_i)$  minus the time elapsed between the arrivals of  $f(N_i)$  and  $p(N_{i-1})$ , i.e.  $(a_{p(N_{i-1})}^{N_i} - a_{f(N_i)}^{N_i})$ . On the example in Figure 4, the parts which have to be considered are the transmission of packet 3 in node  $e3$ , the time elapsed between the arrival of packet 3 and the end of processing of packet 4 in node  $S2$ , the time elapsed between the arrival of packet 4 and the end of processing of packet 5 in node  $S3$ . These parts are shown by thick lines on top of the packets in Figure 4. Their durations are  $40 \mu s$  for  $bp^{e3}$ ,  $4 \mu s$  for  $bp^{S2}$  and  $49 \mu s$  for  $bp^{S3}$ . Thus, the starting time

of packet 3 in node  $S3$  on the example in Figure 4 is:

$$40 + 4 + 49 + (2 \times 16) = 125 \mu s$$

It has been shown [9] that the latest starting time of a packet  $m$  in its last node is reached when  $(a_{p(N_{i-1})}^{N_i} - a_{f(N_i)}^{N_i}) = 0$  for every node  $N_i$  on the path of  $m$ . It comes to postpone the arrival time of every packet joining the path of  $m$  in the node  $N_i$  in order to maximize the waiting time of  $m$  in  $N_i$ .

The result of this postponing on the example in Figure 4 is illustrated in Figure 5. The arrival time of packet 4 at node  $S2$  is postponed to the arrival time of packet 3 at node  $S2$ . In node  $S3$ , packet 5 is postponed in order to arrive between packet 4 and 3, and packet 1 is postponed in order to arrive between  $a_3^{S3}$ , packet 3 arrival time on node  $S3$  and  $s_3^{S3}$ , the transmission time of packet 3 in node  $S3$ .

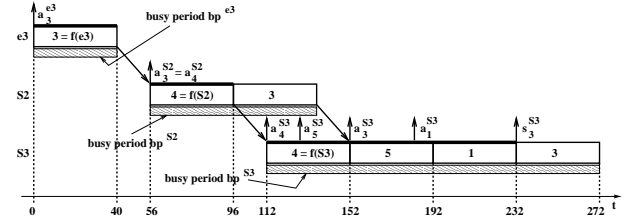


Figure 5. Latest starting time of packet 3

Then, the worst case end-to-end delay of a packet is obtained by adding its latest starting time on its last visited node and its processing time in this last node. For packet 3 in Figure 5, this worst case end-to-end delay is  $232 + 40 = 272 \mu s$ . More precisely, this delay includes the transmission times of packet 3 on node  $e3$ , packet 4 on node  $S2$  and packets 4, 1, 5 and 3 on node  $S3$ . On this example, it can be seen that packets 3 and 4 are counted twice. Actually, it has been shown [9] that exactly one packet has to be counted twice in each node, except the slowest one. In the context of the AFDX, all the nodes work at the same speed. Thus, the slowest node can be arbitrarily chosen as the last one. In the example in Figure 5, packet 3 and 4 are respectively counted twice in nodes  $e3$  and  $S2$ . Packet 3 is the longest one transmitted in nodes  $e3$  and  $S2$ , while packet 4 is the longest one transmitted in node  $S2$  and  $S3$ .

### 3.2.2 The non-preemption effect

We now illustrate the non-preemption effect by studying the end-to-end delay of a packet from VL  $v1$  on the same AFDX configuration. In switch  $S1$ , packet 1 has the highest priority. Thus it cannot be delayed by more than one lower priority packet. This packet has started transmission an arbitrarily small instant  $\epsilon$  before  $a_1^{S1}$  and it cannot be interrupted, due to the non-preemptive characteristic of AFDX. Here, this lower priority packet is packet 2 from VL  $v2$ . The same scenario happens in switch  $S3$ , where  $v1$  is the highest priority flow, but is delayed by one packet

of VL  $v5$ . Thus, the worst case end to end delay of  $v1$  is :  $3 \times C_1 + C_2 + C_5 - 2 \times \epsilon + 2 \times L = 5 \times 40 - 2 \times \epsilon + 2 \times 16 = 232 \mu s$ .

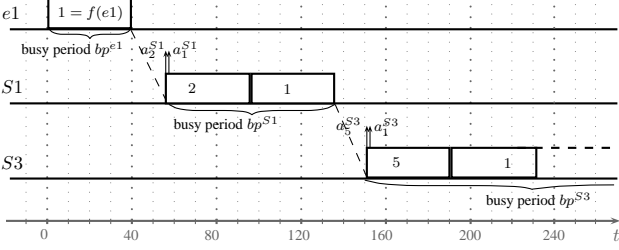


Figure 6. Latest starting time of packet 1

### 3.2.3 Non optimality in the context of an AFDX

In the context of an AFDX network, it is not always possible to find a scheduling which cancels the term  $(a_{p(N_{i-1})}^{N_i} - a_{f(N_i)}^{N_i})$  for every node  $N_i$ , as proposed in [9]. Let us consider VL  $v5$  of the example depicted in Figure 3.  $bp^{S3}$  is the busy period of level corresponding to the priority of packet 5. In order to maximize the delay of packet 5 in  $bp^{S3}$ , the arrival time of packets 3 and 4 in  $S3$  have to be as large as possible, but not larger than the arrival time of packet 5 in node  $S3$ , because of the FIFO scheduling policy of flows with the same fixed priority:

$$a_3^{S3} \leq a_5^{S3} \text{ and } a_4^{S3} \leq a_5^{S3} \quad (1)$$

Since the two packets come from the same link, they are already serialized:

$$|a_3^{S3} - a_4^{S3}| \geq C = 40 \mu s \quad (2)$$

Without loss of generality, let us consider that packet 3 arrives before packet 4. From (1), we have:

$$a_4^{S3} = a_5^{S3} \quad (3)$$

From (2) and (3), we have:

$$a_3^{S3} = a_5^{S3} - 40 \mu s \quad (4)$$

The resulting worst-case scheduling is depicted in Figure 7.  $p(e5)$  is packet 5 and  $f(S3)$  is packet 3. From (4), we have  $(a_{p(e5)}^{S3} - a_{f(S3)}^{S3}) \geq 40 \mu s$  for any possible scheduling. Thus, considering that  $(a_{p(N_{i-1})}^{N_i} - a_{f(N_i)}^{N_i}) = 0$  for every node  $N_i$  is a pessimistic assumption in the context of the AFDX.

## 3.3 Optimization of the Trajectory approach computation

### 3.3.1 Basic computation

The computation of the worst-case end-to-end delay of a packet of a flow  $\tau_i$  has been formalized in [9]. In our context, all the nodes work at the same rate and the jitter

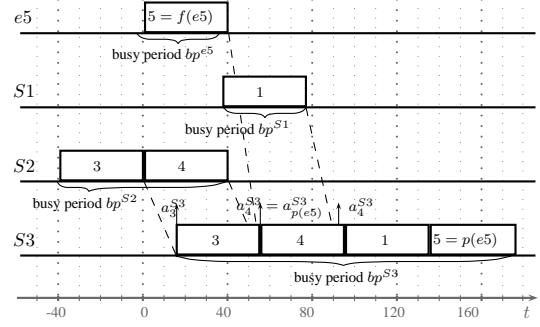


Figure 7. Latest starting time of VL  $v5$

in each emitting node is null. Thus, the worst case end-to-end response time of any flow  $\tau_i$  is bounded by:

$$R_i = \max_{t \geq 0} (W_{i,t}^{last_i} + C_i - t)$$

$last_i$  is the last visited node of flow  $\tau_i$  and  $W_{i,t}^{last_i}$  is a bound on the latest starting time of a packet  $m$  generated at time  $t$  on its last visited node. The definition of  $W_{i,t}^{last_i}$  given in [9] becomes:

$$W_{i,t}^{last_i} = \sum_{\substack{j \in sp_i \cup \{i\} \\ \mathcal{P}_j \cap \mathcal{P}_i \neq \emptyset}} \left( 1 + \left\lfloor \frac{t + A_{i,j}}{T_j} \right\rfloor \right) \cdot C_j \quad (5)$$

$$+ \sum_{\substack{j \in hp_i \\ \mathcal{P}_j \cap \mathcal{P}_i \neq \emptyset}} \left( 1 + \left\lfloor \frac{W_{i,t}^{last_i,j} + B_{i,j}}{T_j} \right\rfloor \right) \cdot C_j \quad (6)$$

$$+ \sum_{\substack{h \in \mathcal{P}_i \\ h \neq last_i}} \left( \max_{\substack{j \in hp_i \cup sp_i \cup \{i\} \\ h \in \mathcal{P}_j}} \{C_j\} \right) \quad (7)$$

$$+ (|\mathcal{P}_i| - 1) \cdot L_{max} \quad (8)$$

$$+ \sum_{N_h \in \mathcal{P}_i} \delta_i^h \quad (9)$$

$$- \sum_{\substack{N_h \in \mathcal{P}_i \\ N_h \neq first_i}} \Delta_{N_h} \quad (10)$$

$$- C_i \quad (11)$$

Where

$$\Delta_{N_h} = a_{p(N_{h-1})}^{N_h} - a_{f(N_h)}^{N_h} \quad (12)$$

Term (5) corresponds to the processing time of packets from flows, crossing the flow  $\tau_i$ , with a fixed priority level equal to this of  $\tau_i$  and transmitted in the same busy period as  $m$ .  $A_{i,j}$  integrates the maximum jitter of packets from  $\tau_i$  and  $\tau_j$  on their first shared output port.

Term (6) is similar to the previous one, but concerns the packets from flows with a fixed priority level higher than this of  $\tau_i$ .  $B_{i,j}$  integrates the maximum jitter of packets from  $\tau_i$  and  $\tau_j$  on their last shared output port. Higher priority packet can overtake  $m$  until its effective transmission in their last shared node ( $W_{i,t}^{last_i,j}$ ). The amount of packet

that can delay the departure of packet  $m$  in its last node has thus to be computed iteratively.

Term (7) is the processing time of the longest packet for each node of path  $\mathcal{P}_i$ , except the last one. It represents the packets which have to be counted twice, as explained before.

Term (8) corresponds to the sum of switching delay.

Term (9) corresponds to the maximum delay due to the non preemption of packets with a fixed priority lower than this of  $\tau_i$ . In each node  $h$ , it is the transmission time of the biggest lower priority packet of a flow  $\tau_j$  crossing flow  $\tau_i$  in this node. It is denoted  $\delta_i^h$ .

Term (10) sums for each node  $N_h$  in  $\mathcal{P}_i$  the duration between the beginning of the busy period and the arrival of the first packet coming from the preceding node in  $\mathcal{P}_i$ , i.e.  $N_{h-1}$ . This term is null in the context of [9].

$C_i$  is subtracted, because  $W_{i,t}^{last_i}$  is the latest starting time and not the ending time of the packet from  $\tau_i$  on its last node.

Solving  $R_i = \max(W_{i,t}^{last_i} + C_i - t)$  comes to find the maximum vertical deviation between the function  $t \mapsto W_{i,t}^{last_i} + C_i$  and the identity function ( $t \mapsto t$ ).

This computation is illustrated on VL  $v_3$  of Figure 3. As there is no flow with a fixed priority lower than this of  $v_3$ , the term (9) is null. For this computation, we also consider that Term (10) is null:

$$\begin{aligned}
W_{3,t}^{S_3} + C_3 &= \left(1 + \left\lfloor \frac{W_{3,t}^{S_3} + B_{3,1}}{T_1} \right\rfloor\right) \cdot C_1 + \left(1 + \left\lfloor \frac{t + A_{3,4}}{T_4} \right\rfloor\right) \cdot C_4 \\
&+ \left(1 + \left\lfloor \frac{t + A_{3,5}}{T_5} \right\rfloor\right) \cdot C_5 + \left(1 + \left\lfloor \frac{t}{T_3 + A_{3,3}} \right\rfloor\right) \cdot C_3 \\
&+ \max_{j \text{ with } S_3 \in \mathcal{P}_j} \{C_j\} + \max_{j \text{ with } S_2 \in \mathcal{P}_j} \{C_j\} - C_3 \\
&+ (|\mathcal{P}_3| - 1) \cdot L_{max} + C_3 \\
&= \left(1 + \left\lfloor \frac{W_{3,t}^{S_3} + 40}{4000} \right\rfloor\right) \cdot 40 + \left(1 + \left\lfloor \frac{t + 40}{4000} \right\rfloor\right) \cdot 40 \\
&+ \left(1 + \left\lfloor \frac{t + 40}{4000} \right\rfloor\right) \cdot 40 + \left(1 + \left\lfloor \frac{t}{4000} \right\rfloor\right) \cdot 40 \\
&+ 40 + 40 - 40 + (3 - 1) \times 16 + 40 \\
&= 272 + 120 \left\lfloor \frac{t + 40}{4000} \right\rfloor + 40 \left\lfloor \frac{W_{3,t}^{S_3} + 40}{4000} \right\rfloor
\end{aligned}$$

The upper-bound on the end-to-end delay is reached for  $t = 0$  and  $R_5^{S_3} = 272\mu s$ .

### 3.3.2 Optimization of the computation

The optimization of this computation in the context of the AFDX concerns Term (10).

Indeed, it has been shown in Section 3.2 that, for some VLs, there exists no scheduling leading to  $\Delta_{N_h} = 0, \forall N_h \in \mathcal{P}_i$ . In the following, we describe the computation of a lower bound on  $\Delta_{N_h}, \forall N_h \in \mathcal{P}_i$  and we prove its correctness.

The value of  $\Delta_{N_h}$  is illustrated in Figure 8.

The packet  $m$  of flow  $\tau_i$  under study is sent on the output link  $OP^h$  in a busy period  $bp^{N_h}$ . The packets which compose  $bp^{N_h}$  in the worst case scenario are determined thanks to terms (5), (6) and (9). These packets are grouped by input link.  $IP_0^h$  is the input link of  $\tau_i$ , while  $IP_x^h$  ( $1 \leq x \leq k_h$ ) are the other input links. Sequence  $seq_x^h$

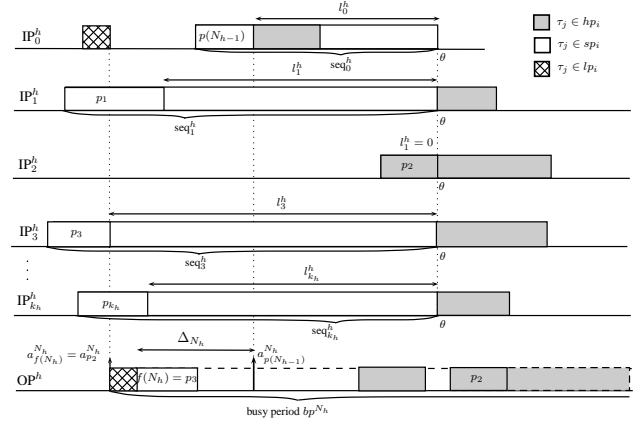


Figure 8. Illustration of  $\Delta_{N_h}$

( $0 \leq x \leq k_h$ ) contains the packets of  $bp^{N_h}$  coming from  $IP_x^h$ .

As defined in (12),  $\Delta_{N_h}$  is the delay between the earliest arrival of a packet of  $bp^{N_h}$  (i.e. the beginning of  $bp^{N_h}$ ) and the arrival of the first packet of  $bp^{N_h}$  coming from  $IP_0^h$ . In Figure 8,  $\Delta_{N_h}$  is the difference between the arrival of  $p_3$  and the arrival of  $p(N_{h-1})$  minus the transmission time of a lower priority packet.

In order to maximize the delay of packet  $m$  in node  $N_h$ , sequences of packets having a fixed priority equal to this of  $m$  are postponed so that their last packet arrives at the same time  $\theta$  as packet  $m$ . Indeed, these packets cannot delay packet  $m$  if they arrive after time  $\theta$ . This construction is a generalization of the Trajectory approach presented in [9]: instead of postponing individually each packet, sequences of already serialized packets are postponed.

Let us consider packets with a higher fixed priority ( $\tau_j \in hp_i$ ). These packets can delay packet  $m$  even if they arrive on node  $h$  after time  $\theta$ . To minimize the value of  $\Delta_h$ , we include them in  $seq_0^h$ , so that  $p(N_{h-1})$  arrives earlier, and we remove them from  $seq_x^h$  ( $1 \leq x \leq k_h$ ) by postponing them after time  $\theta$ , so that  $f(N_h)$  arrives later. When there is no flow from  $sp_i$  in an input port, the higher priority packets are postponed so that the first one arrives at time  $\theta$ . In Figure 8, all the higher priority packets coming from  $IP_2^h$  can delay packet  $m$  if we consider that  $p_2$  is the largest one and the other packets follow by decreasing size. The sequences of packets from  $IP_1^h$  and  $IP_3^h$  are shortened by placing all the higher priority packets after time  $\theta$ , and they are still served before packet  $m$  because of the amount of backlog in the output port.

We also consider the case of the lower priority packet generating the non preemption effect, corresponding to the Term (9). By definition, it arrives just before  $f(N_h)$ . Then, the transmission time of this packet delays the start of the transmission of packet  $p(N_{h-1})$  and also reduces the value of  $\Delta_h$ . In Figure 8, this packet arrives from  $IP_0^h$  just before time  $a_{f(N_h)}^{N_h}$ . Thus, it delays the start of the transmission of packet 3 in the output port of node  $h$ .

The latest starting time of  $m$  in its last node is maxi-

mized when  $\sum_{N_h \in \mathcal{P}_i} (\Delta_{N_h})$  is minimized. It comes to determine the lower bound of each term  $\Delta_{N_h}$  of the sum.

From (12), it is obvious that the minimum value of  $\Delta_{N_h}$  is obtained by minimizing  $a_{p(N_{h-1})}^{N_h}$  and maximizing  $a_{f(N_h)}^{N_h}$ .

Let us define  $l_x^h$  ( $0 \leq x \leq k_h$ ) as the duration of sequence  $seq_x^h$  without its first packet. Then, we have:

$$a_{f(N_h)}^{N_h} = \theta - \max_{1 \leq x \leq k_h} l_x^h \quad (13)$$

$$a_{p(N_{h-1})}^{N_h} = \theta - l_0^h \quad (14)$$

Consequently, minimizing  $a_{p(N_{h-1})}^{N_h}$  comes to maximize  $l_0^h$ . It is obtained when the smallest packet of sequence  $seq_0^h$  is transmitted at the beginning of  $seq_0^h$ .

Similarly, maximizing  $a_{f(N_h)}^{N_h}$  comes to minimize each  $l_x^h$  for  $1 \leq x \leq k_h$ . It is obtained when the largest packet of sequence  $seq_x^h$  is transmitted at the beginning of  $seq_x^h$ , for  $1 \leq x \leq k_h$ .

To summarize,  $\Delta_{N_h}$  is lower bounded by the maximum of 0 and:

$$\max_{1 \leq x \leq k_h} (\min(l_x^h)) - \max(l_0^h) - \max_{y \in \mathcal{L}_i} C_y^h \quad (15)$$

### 3.4 Results on the sample AFDX configuration

The end-to-end delay upper bounds for all the VLs of the configuration in Figure 3 are presented in Table 1. The BT row corresponds to the classical Trajectory approach for FP/FIFO scheduled flows. The OT row gives the enhanced results obtained by applying the grouping optimization. The exact worst case obtained with a model checking tool are also presented in Table 1 for comparison purpose. The network calculus approach is not considered since it has not been applied to the AFDX for FP/FIFO scheduled flows.

VL	EWC ( $\mu s$ )	BT ( $\mu s$ )	OT ( $\mu s$ )
$v1$	232	232	232
$v2$	192	192	192
$v3$	272	272	272
$v4$	272	272	272
$v5$	176	216	176

EWC: exact worst-case

BT: basic Trajectory approach

OT: optimized Trajectory approach

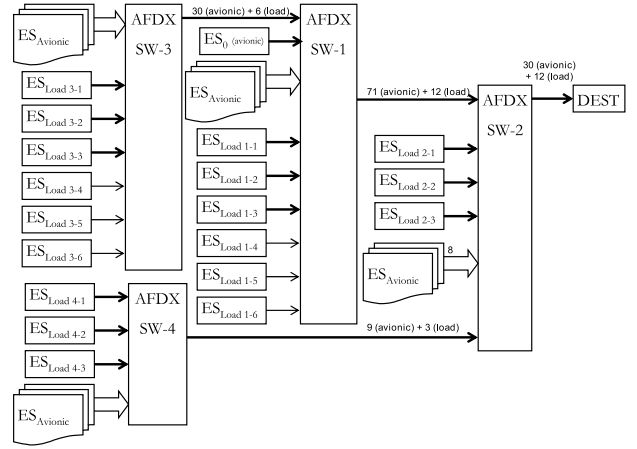
**Table 1. upper end-to-end delays in  $\mu s$**

There are four VLs ( $v1$ ,  $v2$ ,  $v3$  and  $v4$ ) for which the basic Trajectory approach gives the exact worst case. However, for VL  $v5$ , the basic Trajectory approach introduces a  $40\mu s$  pessimism, which is eliminated by the optimization of the computation.

## 4 Illustration on a representative part of an industrial AFDX network

### 4.1 An AFDX network with additional lower priority flows

The results presented in this section are based on the AFDX architecture depicted in Figure 9. This configuration is a representative part of an industrial AFDX network. The part considered in this study includes 21 end systems, four switches and 91 VLs. Each avionics system is distributed on different end systems connected through VLs. We consider the avionics VL connecting end system  $ES_0$  to end system  $ES_{DEST}$ , crossing switches  $SW_1$  and  $SW_2$ , as the reference VL for this study. This flow will be named  $VL_0$ . It has the following parameters:  $BAG = 32ms$  and  $s_{max} = 384bytes$ .



**Figure 9. An AFDX configuration with additional lower priority flows**

In order to evaluate QoS performance, 18 end systems, each emitting one lower priority flow, are added to the reference network. These VLs will increase the load on output ports of the AFDX switches. Thin arrows designate VLs that follow the same path than  $VL_0$  only in  $SW_1$  while bold arrows designate VLs that also follow the same path than  $VL_0$  in  $SW_2$ . These additional VLs all have the same  $s_{max}$  and  $BAG$  parameters, which are set to impose a given load in the output ports of the switches.

All the worst case end-to-end delay results presented in this section have been computed with our Trajectory tool with either FIFO or FP/FIFO scheduling algorithms. The initial worst case end-to-end delay of  $VL_0$  is  $1473\mu s$ . This value is a reference to evaluate the impact on the avionics traffic of the additional load. Similarly, we select a VL from the additional traffic that follows the same path as  $VL_0$ , except for the source end system (which is  $ES_{Load 1-1}$ ). As it is not possible to give a precise figure about the global network load, we take the output port of switch  $SW_1$  as a reference for the load information. As all the additional VLs have identical parameters, the variation of load in switch  $SW_1$  is representative of the global

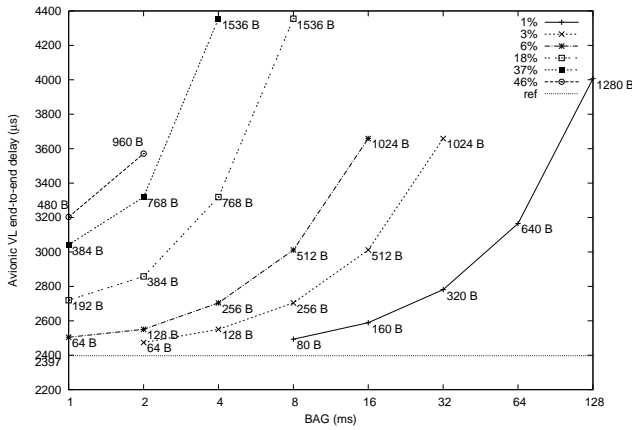


network load variation. The initial load in the output port of switch SW<sub>1</sub> without any additional load is about 13%.

#### 4.2 Impact of a fixed priority strategy

Adding lower priority traffic to the existing avionics traffic to better use available network resource is a promising idea but this efficiency increase should in no case impact the determinism of avionics data flows. The determinism of those flows is closely related to their worst case end-to-end response time. Thus, we have computed these bounds for the reference VL<sub>0</sub> for different additional loads traffic. At first, we evaluate the evolution of response time with a FIFO algorithm. This gives us a reference to measure the interest of a FP/FIFO scheduling solution. The results for FIFO are depicted in Figure 10. Both end-to-end delay analysis are conducted with the computation tool we developed and which features the Trajectory approach optimization presented in Section 3.3.

##### 4.2.1 Mitigating the impact on the existing avionics flows

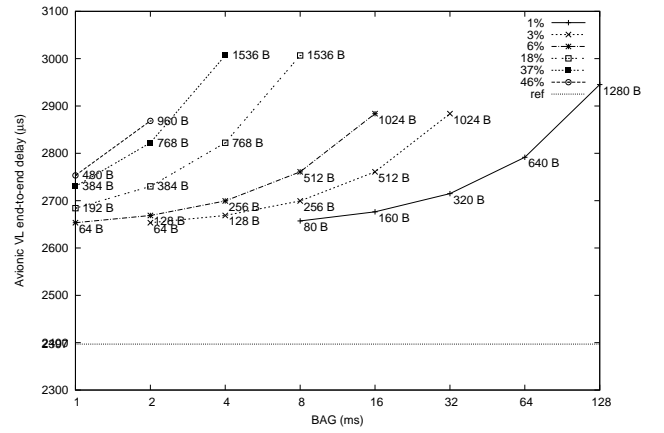


**Figure 10. Impact of an additional load on the end-to-end delay of VL<sub>0</sub> with a FIFO servicing policy**

Each flow generates a load depending on its BAG and  $s_{\max}$  parameters. A given load can correspond to many  $(BAG, s_{\max})$  combinations. If we do not consider the per-frame overhead, the same information can be sent with 80bytes frames every 8ms, with 160bytes frames every 16ms or with 320bytes frames every 32ms, and so on... In AFDX networks, the BAG values are harmonic periods between 1ms and 128ms.  $s_{\max}$  is limited by the standard Ethernet frame size. In Figure 10, curves represent iso-load evolutions. The reference line is the end-to-end response time with a null additional load. Each point corresponds to a  $(BAG, s_{\max})$  couple. The BAG value is given on the horizontal axis and the  $s_{\max}$  value is displayed next to the point.

In the FIFO case, the end-to-end delay of VL<sub>0</sub> is mainly impacted by the  $s_{\max}$  of the additional flows even

at low level loads : the impact of a 37% additional load with  $(BAG = 1ms, s_{\max} = 384B)$  parameters is lower than for a 1% load and  $(64ms, 640B)$  parameters. Generally, the impact goes lower with lower  $s_{\max}$  values. This is because in the worst case and for relatively low loads, an avionics packet will have to wait in an output port during the transmission of packets from other flows, which directly depends on their size. Moreover, using only minimal packet size generates a high amount of overhead. We conclude from this figure that FIFO cannot cope with such a traffic increase. We want to see if the avionics flows can be preserved with a fixed priority QoS policy. The results are depicted in Figure 11, which gives the same type of information on worst case end-to-end delays of VL<sub>0</sub> that Figure 10.



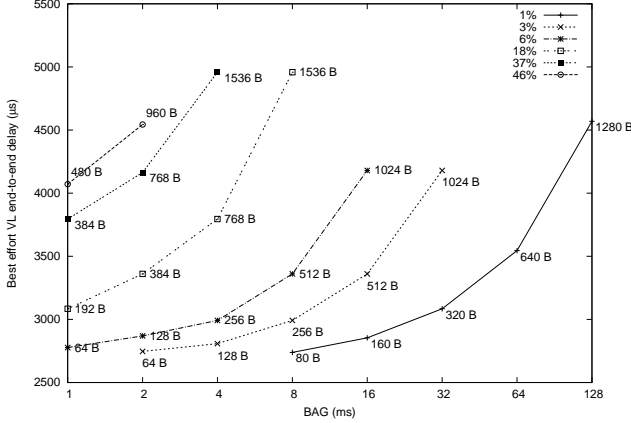
**Figure 11. Impact of an additional load on the end-to-end delay of VL<sub>0</sub> with a FP/FIFO servicing policy**

Not surprisingly, the results for VL<sub>0</sub> with SP/FIFO are much better than with FIFO. Although there is still an impact on avionics worst case end-to-end response times with increasing packet size, the increase is much more limited. The delay increase with 1024B packets is up to 20% lower than in FIFO. An avionics VL is still delayed by lower priority level flows, but only due to the non pre-emption effect, which directly depends on the size of the lower priority packets.

The only packets of lower priority that can delay an avionics packet in a switch output port is a packet that is already being served. As the avionics load is constant, the end-to-end delay bound only depends on the  $s_{\max}$  parameter of the lower priority flows. Indeed, we can observe that all the point with a similar  $s_{\max}$  value are aligned and correspond to a given worst case end-to-end delay for VL<sub>0</sub>. This means that the impact on avionics traffic can easily be contained, simply by limiting the  $s_{\max}$  parameter of the additional traffic, independently of the load. This requirement is easy to specify and guarantees a certain scalability for future increase of the additional load.

#### 4.2.2 Impact of BAG and $s_{max}$ on lower priority flows

For the lower priority flows, FP/FIFO and FIFO results are quite similar: as these flows represent the major part of the traffic, the impact of the higher priority avionics flows is discernible only for very low loads. The same amount of loads as previously are presented in Figure 12.



**Figure 12. Impact of an additional load on the end-to-end delay of  $VL_{Load1-1}$  with a FP/FIFO servicing policy**

The worst case end-to-end delays decreases with the  $s_{max}$  value because partly because of the shorter transmission time of shorter frames in output buffers and because the other additional flows send shorter packets that create less congestion in the output ports. But this effects is reversed with small values of BAG and with high loads: a lower priority packet can be delayed from more than one packet from a same source. The busy periods becomes much longer and the Trajectory upper bound increases. This phenomenon is clearly visible for a 6% load and a 19% load with a  $BAG = 1ms$ .

#### 4.3 Gain of the grouping optimization

##### 4.3.1 Gain for high priority $VL_0$

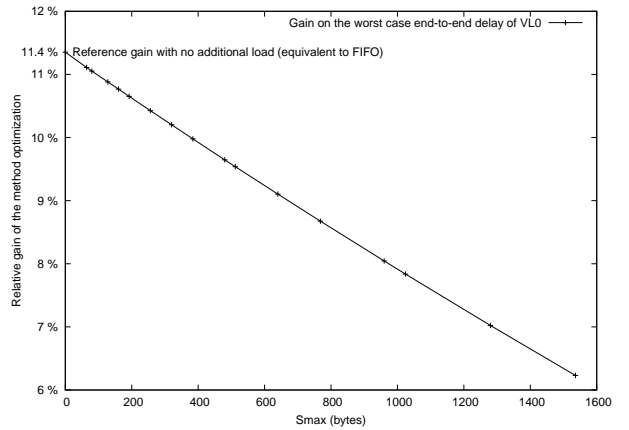
We study the gain of the grouping optimization for the avionics flow  $VL_0$ . Table 2 gives the relative gain of the optimization for different additional lower priority traffic loads. The empty cells correspond to impossible combinations of BAG and  $s_{max}$  values (because of the AFDX frame minimal and maximal size). The 0% row corresponds to the initial avionics configuration without additional traffic. In this case, the gain of the grouping optimization is 11.4%. With additional load, the gain of the optimization is still existing, but decreases with the load increases. In fact, we show in Figure 13 that the evolution of the gain is directly linked to the  $s_{max}$  parameter of the additional load. Indeed, in the  $\Delta_h$  computation, the only impact of lower priority flows is the transmission time of one lower priority packet, due to the non preemption ef-

BAG	0% load	1% load	3% load	6% load	18% load	37% load	45% load
1 ms		-	-	11.1%	10.7%	10.0%	9.6%
2 ms		-	11.1%	10.9%	10.0%	8.7%	8.0%
4 ms		-	10.9%	10.4%	8.7%	6.2%	-
8 ms		11.1%	10.4%	9.5%	-	-	-
16 ms		10.8%	9.5%	7.8%	-	-	-
32 ms		10.2%	7.8%	-	-	-	-
64 ms		9.1%	-	-	-	-	-
128 ms		7.0%	-	-	-	-	-

**Table 2. Gain of the grouping optimization on  $VL_0$  worst case end-to-end delay**

fect.

The worst case end-to-end delay without the grouping optimization for  $VL_0$  is  $2704\mu s$ . With the optimization, this bound falls to  $2397\mu s$ . This represents the initial 11.4% gain. With a 16% additional load of 1536 byte packets and a  $16ms$  BAG, the bound without optimization raises up to  $2950\mu s$ . This raise corresponds to the  $246\mu s$  needed to transmit two 1536 bytes packets due to the non preemption effect. With the grouping optimization, the bound raises up to  $2766\mu s$ . This  $369\mu s$  raise corresponds to the transmission time of three 1536 bytes packets ( $3 \times 1536 \times 8/100 = 368.64\mu s$ ). Two of them correspond to the non preemption effect, as previously. The third packet is removed from  $\Delta_h$  in the worst case end-to-end delay computation of  $VL_0$  in node SW-1. Indeed, this is the node where the grouping phenomenon occurs:  $VL_0$  meets 30 already serialized avionics flows coming from switch SW-3. In the worst case, the transmission time of a packet of lower priority has to be subtracted from  $\Delta_h$ . This is where the third occurrence of a 1536 bytes lower priority packet has to be counted.



**Figure 13. Gain of the grouping optimization on  $VL_0$  worst case end-to-end delay**

##### 4.3.2 Gain for lower priority $VL_{Load1-1}$

In Table 3, the gain of the optimization for each amount of additional traffic is summarized. The gain is higher for larger packet sizes (for each row, the packet size is

proportional to the BAG). The delay introduced by the six lower priority serialized flows from switch SW-3 that cross the path of  $VL_{Load1-1}$  in switch SW-1 is reduced to only one with the serialization optimization. The five packet gain is thus higher for larger packet sizes. If we consider the configuration in the case of a 128ms BAG and a 1280 bytes packet size (last row of the first column), the optimization gain is  $631.76\mu s$ . This corresponds to the transmission time of five 1280 bytes packets ( $5 \times 1280 \times 8/100 = 512\mu s$ ) and the transmission time of one higher priority packet of 1497 bytes ( $119.76\mu s$ ). This packets is from  $VL_{567}$ . Because of the small BAG of this VL (2ms), more than one packet from this flow can delay a packet from  $VL_{Load1-1}$ . In fact, the serialization optimization reduces this phenomenon which happens with high loads and low BAG values. This explains also the upturn in the two last rows of Table 3 for the smallest BAG value.

BAG	1% load	3% load	6% load	18% load	37% load	45% load
1	-	-	5.0%	6.1%	15.4%	16.8%
2	-	5.1%	5.7%	7.9%	10.0%	10.8%
4	-	5.8%	7.1%	12.6%	20.2%	-
8	5.3%	7.1%	9.3%	14.5%	-	-
16	6.2%	9.3%	12.3%	-	-	-
32	7.7%	12.3%	-	-	-	-
64	10.3%	-	-	-	-	-
128	13.5%	-	-	-	-	-

**Table 3. Average gain of the grouping optimization on  $VL_{Load1-1}$  worst case end-to-end delay**

## 5 Conclusion

In this paper, we first demonstrated that the Trajectory approach with SP/FIFO scheduling is able to guarantee worst case end-to-end delays for AFDX networks with static priority flows differentiation QoS mechanisms.

Then we showed how this approach can be enhanced in the AFDX context thanks to the grouping technique. This improvement allows to compute tighter upper-bounds on end-to-end delays. It has been illustrated on a sample AFDX configuration.

Then, we analyzed the impact of additional low priority traffic on a representative part of a industrial AFDX network. We showed that the impact of low priority flows can be upper bounded per switch by the transmission time of the biggest lower priority packet (non preemption effect). Moreover, as the load induced by avionics flows is low, we conclude that the impact of SP/FIFO policy on lower priority flows is limited.

## References

- [1] ARINC Specification 664: Aircraft Data Network, Parts 1,2,7. Technical report, Aeronotical Radio Inc., 2002-2005.
- [2] H. Bauer, J.-L. Scharbarg, and C. Fraboul. Applying and optimizing trajectory approach for performance evaluation

- of AFDX avionics network. In *Proc. of the 14th International Conference on Emerging Technologies and Factory Automation*, Mallorca, Spain, September 2009. IEEE.
- [3] H. Charara, J.-L. Scharbarg, J. Ermont, and C. Fraboul. Methods for bounding end-to-end delays on an AFDX network. In *Proceedings of the 18th ECRTS*, Dresden, Germany, July 2006.
- [4] F. Frances, C. Fraboul, and J. Grieu. Using network calculus to optimize the AFDX network. In *Proceedings of ERTS*, Toulouse, France, 2006.
- [5] J. Grieu. *Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques*. PhD thesis, INP-ENSEEIH, France, September 2004.
- [6] J. Jasperneite, P. Neumann, M. Theis, and K. Watson. Deterministic Real-Time Communication with Switched Ethernet. In *Proceedings of the 4th IEEE International Workshop on Factory Communication Systems*, pages 11–18, Västerås, Sweden, August 2002. IEEE Press.
- [7] S. Martin. *Maîtrise de la dimension temporelle de la qualité de service dans les réseaux*. PhD thesis, Université Paris XII, July 2004.
- [8] S. Martin and P. Minet. Schedulability analysis of flows scheduled with fifo: application to the expedited forwarding class. *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, pages 8 pp.–, April 2006.
- [9] S. Martin and P. Minet. Worst case end-to-end response times of flows scheduled with fp/fifo. In *ICNICONSMCL '06: Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, page 54. IEEE Computer Society, 2006.

## Notations

- $\tau_i$ : one flow on the network
- $\mathcal{P}_i$ : path of flow  $\tau_i$
- $hp_i$ : set of flows having a fixed priority strictly higher than this of  $\tau_i$
- $sp_i$ : set of flows having a fixed priority equal to this of  $\tau_i$
- $lp_i$ : set of flows having a fixed priority strictly lower than this of  $\tau_i$
- $T_i$ : minimum delay between two packets of  $\tau_i$
- $J_i$ : maximum release jitter of  $\tau_i$
- $C_i$ : transmission time of one packet of  $\tau_i$
- $L_{max}$ : maximum switching delay
- $a_m^{N^h}$ : arrival time of packet  $m$  in node  $N^h$
- $bp^{N^h}$ : busy period of node  $N^h$
- $f(N^i)$ : first packet processed in  $bp^{N^i}$
- $p(N^{i-1})$ : first packet processed in  $bp^{N^i}$  and coming from  $N^{i-1}$
- $R_i$ : worst-case end-to-end response time of  $\tau_i$
- $W_{i,t}^{last i}$ : latest starting time of a packet  $m$  generated at  $t$  on its last visited node
- $A_{i,j}$ : maximum relative jitter between  $\tau_i$  and  $\tau_j$
- $B_{i,j}$ : maximum relative jitter between  $\tau_i$  and  $\tau_j$
- $\delta_i^h$ : maximum non-preemption delay for  $\tau_i$  on node  $h$
- $\Delta_{N_h}$ : impact of serialization on node  $N_h$